



HPCC Systems for Social Good – The Safe Havens Workshop

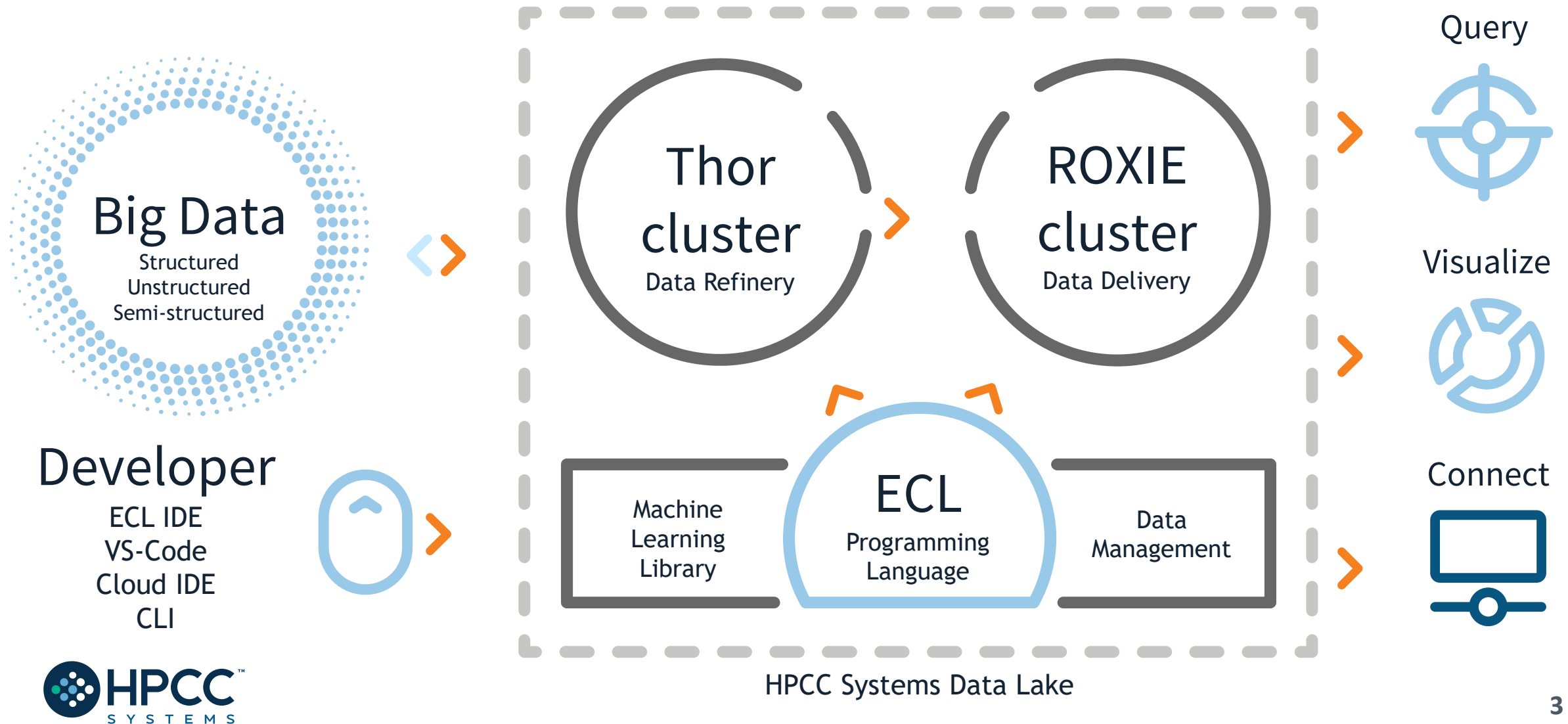
APRIL 2024

Bob Foreman
Software Engineer Lead
LexisNexis Risk Solutions

Welcome to the HPCC Systems Safe Havens Workshop!

- ✓ This workshop uses the HPCC Systems platform and ECL (Enterprise Control Language).
- ✓ HPCC defined is a *distributed data parallel processing* platform.
High Performance Computing Cluster
- ✓ Contains a THOR cluster where the majority of your coding will be done, and a ROXIE cluster to deliver your results.
- ✓ A proven platform for LexisNexis for over 20 years, and open source since 2011!

The HPCC Systems Components



Technology – The Open Source Stack



Thor: Data Refinery Cluster

Extraction, loading, cleansing, transforming, linking and indexing



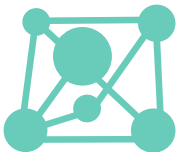
ROXIE: Data Delivery Engine

Rapid data delivery cluster with high-performance online query delivery for big data



Data Management Tools

Data profiling, cleansing, snapshot data updates, consolidation, job scheduling and automation



Machine Learning Library

Linear regression, logistic regression, decision trees and random forests



Connectivity & Third-Party Tools

New plugins to help integrate third party tools with the HPCC Systems platform

An Introduction to ECL

ECL

Enterprise Control
Language



```
IMPORT $, STD, ML;
EXPORT Func(UNSIGNED C, UNSIGNED2 Dist, UNSIGNED size, STRING Fld, REAL Parm1=0, REAL Parm2=0, REAL Parm3=0) := MODULE
  SHARED Node := STD.system.Thorlib.Node()+1;
  SHARED PersistPrefix := $.Parms.PersistPrefix;
  SHARED TotalRecs := $.Parms.RecCnt*CLUSTER_SIZE;
  SHARED UIDval := IF(C=1, node, node + ((C-1)*CLUSTER_SIZE));
  SHARED BOOLEAN IsRandFile := $.Parms.Randomness = $.ut.RandomSrc.file;
  SHARED Normal := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal(Parm1, Parm2),
      ML.Distribution.Normal(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'NormalDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Normal2 := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Normal2(Parm1, Parm2),
      ML.Distribution.Normal2(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'Normal2DistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED Uniform := FUNCTION
    Thisdist := IF(Parm3=0,
      ML.Distribution.Uniform(Parm1, Parm2),
      ML.Distribution.Uniform(Parm1, Parm2, Parm3));
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'UniformDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
  SHARED StudentT := FUNCTION
    Thisdist := ML.Distribution.StudentT(Parm1, Parm2);
    RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'StudentTDistInt' + Fld, EXPIRE(1));
    RETURN RetVals;
  END;
END;
```



- Transparent and implicitly parallel programming language
- Both powerful and flexible

- Optimized for data-intensive operations, declarative, non-procedural and dataflow oriented
- Uses intuitive syntax which is modular, reusable, extensible and highly productive

How to do
it



vs.

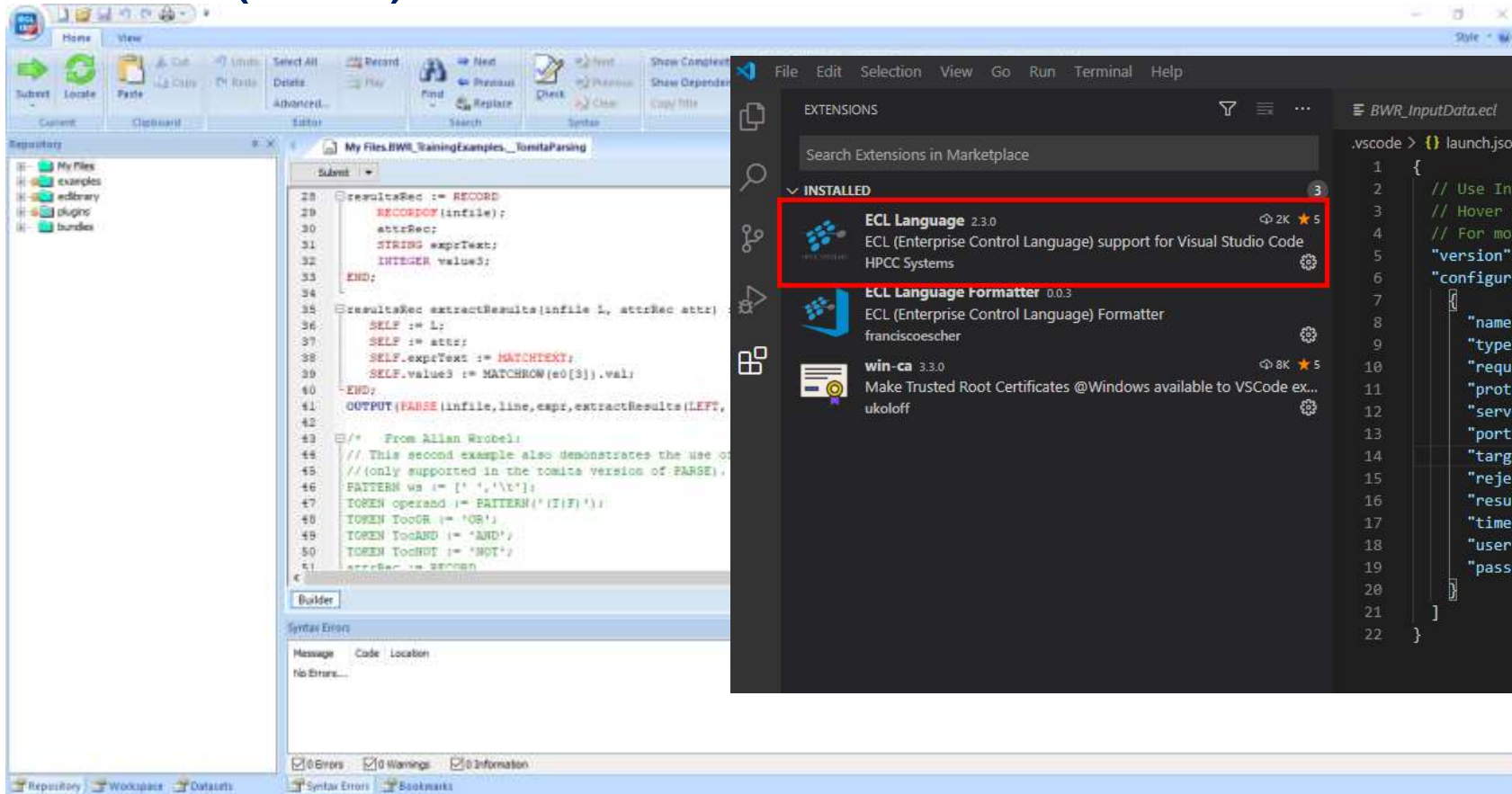


What to
do

Integrated Development Environments

ECL IDE (Win)

Visual Studio Code (Ux/MacOS)



The Playing Field!

HPCC Cluster ECL Watch:

<http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/>

```
"configurations": [  
  {  
    "name": "External",  
    "type": "ecl",  
    "request": "launch",  
    "protocol": "http",  
    "serverAddress": "training.us-hpccsystems-dev.azure.lnrsg.io",  
    "port": 8010,  
    "path": "",  
    "targetCluster": "thor",  
    "rejectUnauthorized": true,  
    "resultLimit": 100,  
    "timeoutSecs": 60,  
    "user": "YourNameHere",  
    "password": ""  
  },  
]
```

Preferences

Configurations: ExternalCluster

Locate New... Delete

Server Editor Colors Results Compiler Other

Server IP: training.us-hpccsystems-dev.azure.lnrsg.io ☐ SSL ☒ Advanced

Topology Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/WsTop

Workunit Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/WsWo

Attribute Server:

Account Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/Ws_Ac

SMC Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/WsSMC

Spray Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/FileSpr

DFU Server: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/WsDfu

ECL Watch URL: http://training.us-hpccsystems-dev.azure.lnrsg.io:8010/esp/fil

Ok Cancel Apply

The Repo!

<https://github.com/hpccsystems-solutions-lab/SafeHavenChallenge>

The screenshot displays the GitHub repository page for **SafeHavenChallenge**, which is a public repository. The repository is currently on the **main** branch, with 1 branch and 0 tags. The repository owner is **bobf2000**, and the last update was "Final Code Update".

The repository structure is as follows:

File/Folder	Commit Message
Code	Final Code Update
Images	ReadMe and Images up
LICENSE	Initial commit
README.md	ReadMe and Images up

The repository is licensed under the **Apache-2.0 license**. The **README** file is highlighted.

The **Code** button is circled in red, and a red arrow points from it to the **Download ZIP** button, which is also circled in red.

IDE Features:



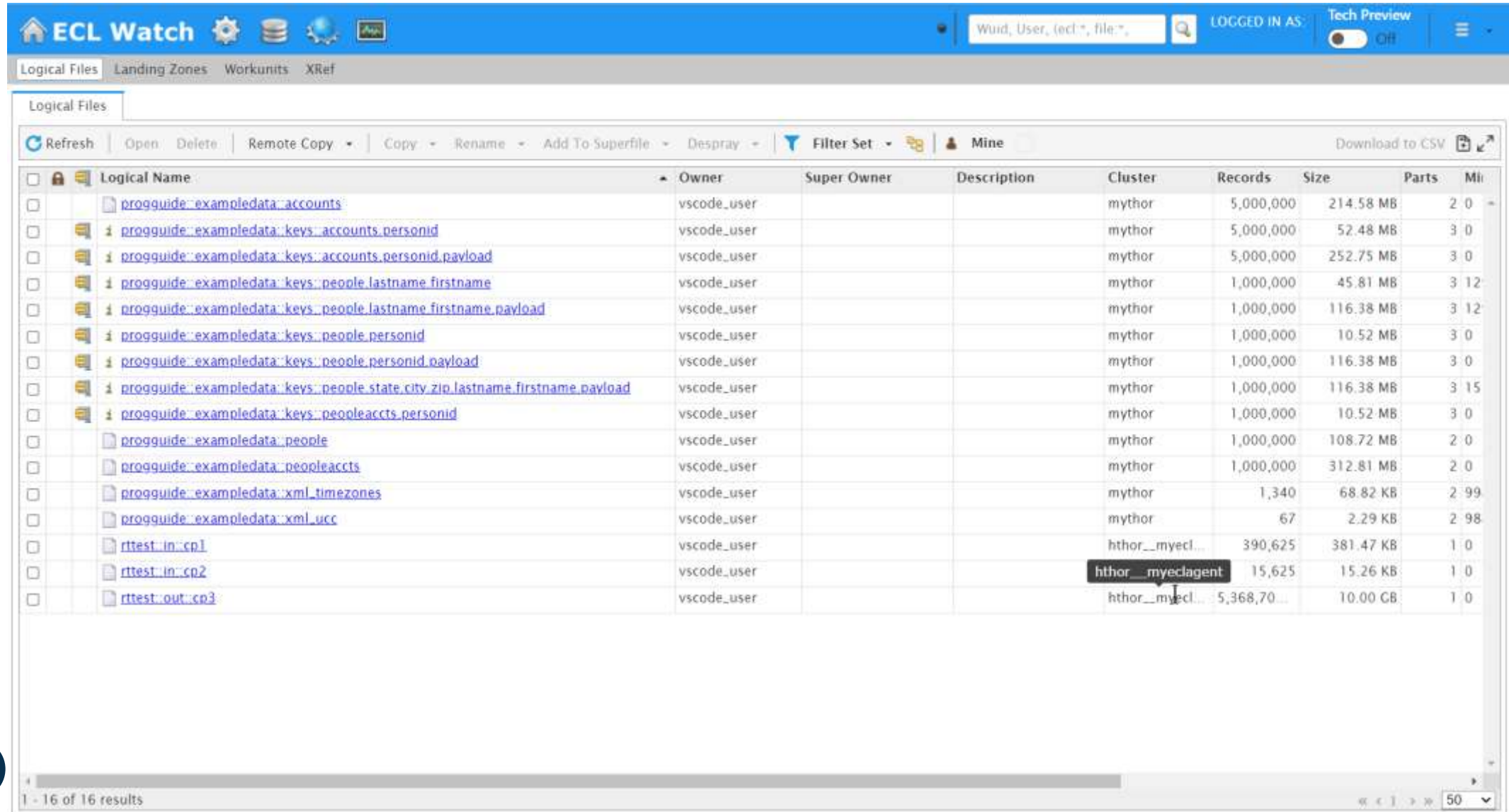
A full-featured GUI for ECL development providing access to the ECL repository and many of the ECL Watch capabilities.

Uses various ESP services via SOAP.

Provides the easiest way to create:

1. Queries into your data, instant results!
2. ECL Definitions to build your queries which:
 - Are created by coding an expression that defines how some calculation or record set derivation is to be done.
 - Once defined, can be used in succeeding ECL definitions.

The ECL Watch (pre-version 9)

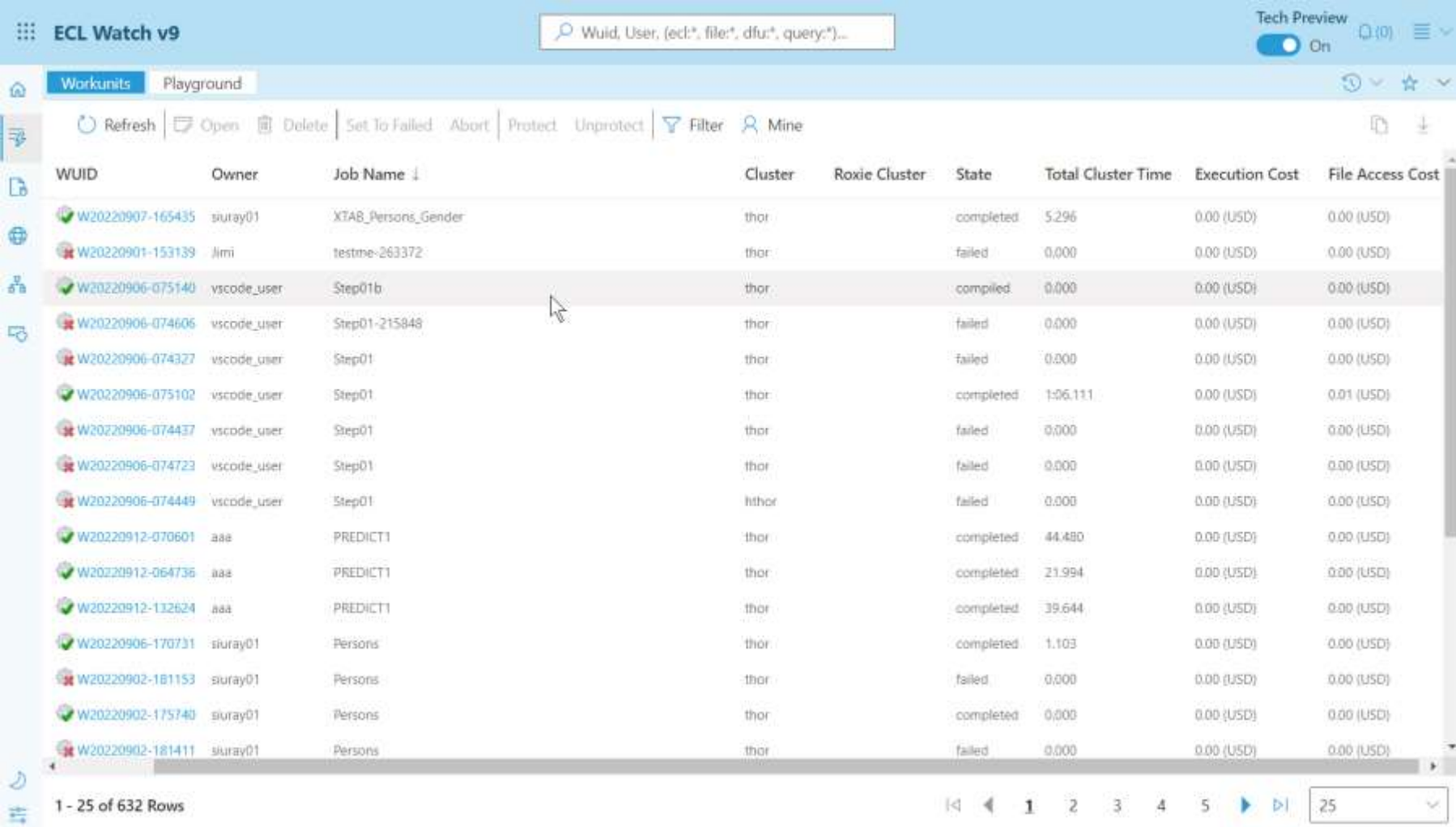


The screenshot shows the ECL Watch application interface. At the top is a blue header bar with the 'ECL Watch' logo, navigation icons, a search bar containing 'Wuid, User, (ecl *, file:*)', a 'LOGGED IN AS' indicator, a 'Tech Preview' toggle set to 'Off', and a menu icon. Below the header is a grey navigation bar with tabs for 'Logical Files', 'Landing Zones', 'Workunits', and 'XRef'. The 'Logical Files' tab is active, displaying a table of logical files. Above the table is a toolbar with buttons for 'Refresh', 'Open', 'Delete', 'Remote Copy', 'Copy', 'Rename', 'Add To Superfile', 'Despray', 'Filter Set', 'Mine', and a 'Download to CSV' button. The table has columns for 'Logical Name', 'Owner', 'Super Owner', 'Description', 'Cluster', 'Records', 'Size', 'Parts', and 'Mi'. It lists 16 logical files, including various 'proqguide::exampledata' and 'rttest' files. The status bar at the bottom indicates '1 - 16 of 16 results' and a page number '50'.

Logical Name	Owner	Super Owner	Description	Cluster	Records	Size	Parts	Mi
proqguide::exampledata::accounts	vscode_user			mythor	5,000,000	214.58 MB	2 0	-
proqguide::exampledata::keys::accounts.personid	vscode_user			mythor	5,000,000	52.48 MB	3 0	
proqguide::exampledata::keys::accounts.personid.payload	vscode_user			mythor	5,000,000	252.75 MB	3 0	
proqguide::exampledata::keys::people.lastname.firstname	vscode_user			mythor	1,000,000	45.81 MB	3 12	
proqguide::exampledata::keys::people.lastname.firstname.payload	vscode_user			mythor	1,000,000	116.38 MB	3 12	
proqguide::exampledata::keys::people.personid	vscode_user			mythor	1,000,000	10.52 MB	3 0	
proqguide::exampledata::keys::people.personid.payload	vscode_user			mythor	1,000,000	116.38 MB	3 0	
proqguide::exampledata::keys::people.state.city.zip.lastname.firstname.payload	vscode_user			mythor	1,000,000	116.38 MB	3 15	
proqguide::exampledata::keys::peopleaccts.personid	vscode_user			mythor	1,000,000	10.52 MB	3 0	
proqguide::exampledata::people	vscode_user			mythor	1,000,000	108.72 MB	2 0	
proqguide::exampledata::peopleaccts	vscode_user			mythor	1,000,000	312.81 MB	2 0	
proqguide::exampledata::xml.timezones	vscode_user			mythor	1,340	68.82 KB	2 99	
proqguide::exampledata::xml.ucc	vscode_user			mythor	67	2.29 KB	2 98	
rttest::in::cp1	vscode_user			hthor__myecl...	390,625	381.47 KB	1 0	
rttest::in::cp2	vscode_user			hthor__myeclagent	15,625	15.26 KB	1 0	
rttest::out::cp3	vscode_user			hthor__myecl...	5,368,70	10.00 GB	1 0	



The ECL Watch 9



The screenshot displays the ECL Watch v9 application interface. At the top, there is a search bar with the placeholder text "Wuid, User, (ecl:*, file:*, dfu:*, query:*)...". To the right of the search bar is a "Tech Preview" toggle switch set to "On". Below the search bar, there are tabs for "Workunits" and "Playground". A toolbar contains buttons for "Refresh", "Open", "Delete", "Set To Failed", "Abort", "Protect", "Unprotect", "Filter", and "Mine". The main area is a table with the following columns: WUID, Owner, Job Name, Cluster, Roxie Cluster, State, Total Cluster Time, Execution Cost, and File Access Cost. The table lists 25 rows of workunit data. The status of each workunit is indicated by a green checkmark for "completed" and a red X for "failed". The bottom of the interface shows a pagination bar with "1 - 25 of 632 Rows" and a set of navigation controls including arrows and page numbers 1 through 5, with a dropdown menu currently showing "25".

WUID	Owner	Job Name	Cluster	Roxie Cluster	State	Total Cluster Time	Execution Cost	File Access Cost
W20220907-165435	siuray01	XTAB_Persons_Gender	thor		completed	5.296	0.00 (USD)	0.00 (USD)
W20220901-153139	jlimi	testtime-263372	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-075140	vscode_user	Step01b	thor		compiled	0.000	0.00 (USD)	0.00 (USD)
W20220906-074606	vscode_user	Step01-215848	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074327	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-075102	vscode_user	Step01	thor		completed	1:06.111	0.00 (USD)	0.01 (USD)
W20220906-074437	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074723	vscode_user	Step01	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220906-074449	vscode_user	Step01	lithor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220912-070601	aaa	PREDICT1	thor		completed	44.480	0.00 (USD)	0.00 (USD)
W20220912-064736	aaa	PREDICT1	thor		completed	21.994	0.00 (USD)	0.00 (USD)
W20220912-132624	aaa	PREDICT1	thor		completed	39.644	0.00 (USD)	0.00 (USD)
W20220906-170731	siuray01	Persons	thor		completed	1.103	0.00 (USD)	0.00 (USD)
W20220902-181153	siuray01	Persons	thor		failed	0.000	0.00 (USD)	0.00 (USD)
W20220902-175740	siuray01	Persons	thor		completed	0.000	0.00 (USD)	0.00 (USD)
W20220902-181411	siuray01	Persons	thor		failed	0.000	0.00 (USD)	0.00 (USD)

ECL Watch Features:

A web-based query execution, monitoring and file management interface. It can be accessed via ECL IDE or a web browser.

ECL Watch allows you to:

1. See information about active workunits.
2. Monitor cluster activity.
3. Browse through previously submitted Workunits.
4. See a visual representation of the data flow within the WU, complete with statistics which are updated as the job progresses.
5. Search through files and see information including:
 - Record counts and layouts.
 - Sample records.
 - The status of all system servers whether they are in clusters or not.
6. View log files.
7. Start and stop processes.





ECL Overview

FEBRUARY 2023

Bob Foreman
Software Engineer Lead
LexisNexis Risk Solutions

ECL (Enterprise Control Language)

ECL is a language design to query/manipulate massive data and is used for ETL (Extract, Transform, Load) and data visualization.

Extract

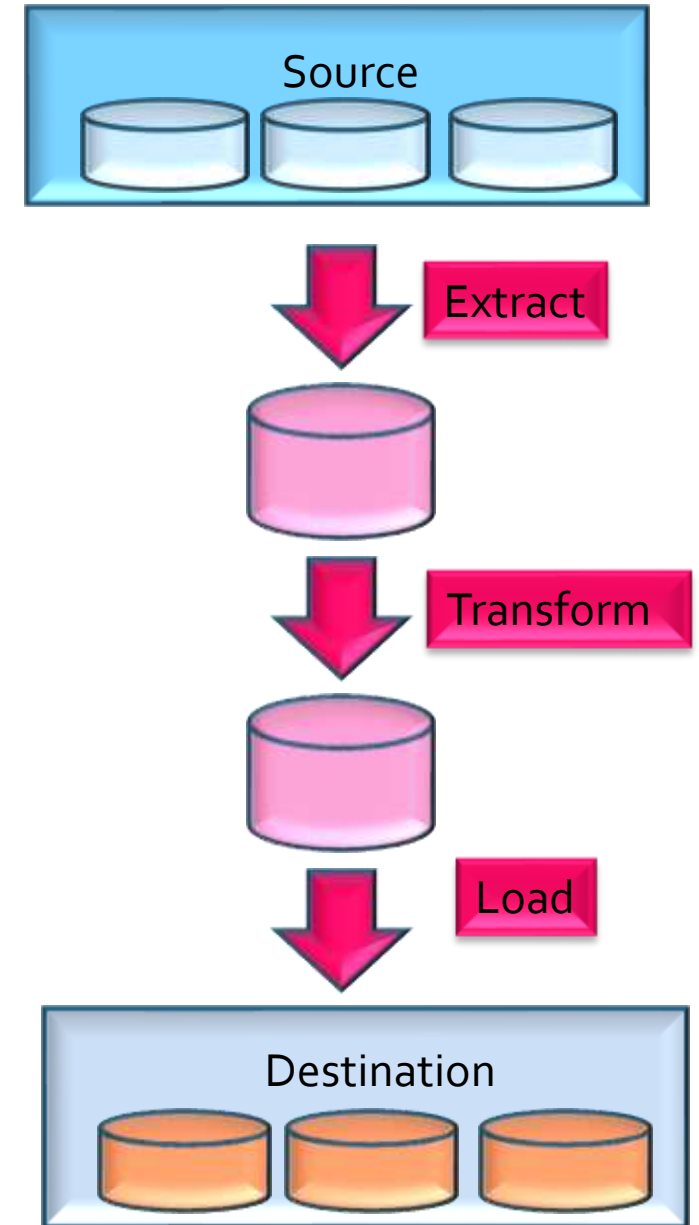
Reading data from different type of datasets

Transform

Formatting/converting data to needed shape

Load

Writing (Delivering) dataset to its target location



Fundamentals of ECL

- ✓ Declarative Language
- ✓ **Not** case-sensitive
- ✓ White space is ignored (Makes your code more readable)

```
// This is a single line comment
/* A block comment */
```
- ✓ *Object.Property* syntax is used to qualify definition scope and disambiguate field references within datasets:
- ✓ *FolderName.Definition* //reference a definition from another module/folder
- ✓ *Dataset.Field* //reference a field in a dataset or record set

Fundamentals of ECL (Continued)

- ✓ Definition operator is **`:=`** “is defined as”
- ✓ Semicolon is line terminator: `num := 12;`
- ✓ Equality test is `=` `valOne = valTwo`
- ✓ Not equal: Use `<>` or `!=`
- ✓ Definitions can be defined only once.
- ✓ *Only those definitions that contribute to a result are compiled and used.*
- ✓ There are no loops as we know them! TRANSFORM and PROJECT is used instead.

Common Data Types

Character

- `STRING[n]`
- `UTF8`
- `UNICODE[_locale][n]`

Numeric

- `INTEGER[n]`
- `UNSIGNED[n]`
- `REAL[n]`
- `DECIMAL<n>[_y]`
- `UDECIMAL<n>[_y]`

Other

- `BOOLEAN`
- `SET OF <type>`
- `RECORD`
- `DATASET`

Usage:

Type Name := default value
`UNSIGNED1 MyNumber := 0;`

Name must start with a letter and can contain letters, numbers and the underscore character.

RECORD Structure

Defines the layout of fields in the dataset, order of the fields should be the same as the dataset.

DATASET

A physical data file. It can be defined in code (inline) or can be read from disk.

Job	Catergory	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

RECORD Structure Example:

```
EXPORT Layout_Company := RECORD
  UNSIGNED  sic_code;
  STRING1   source;
  STRING120 company_name;
  STRING10  prim_range;
  STRING2   predir;
  STRING28  prim_name;
  STRING4   addr_suffix;
  STRING2   postdir;
  STRING5   unit_desig;
  STRING8   sec_range;
  STRING25  city;
  STRING2   state;
  STRING5   zip;
  STRING4   zip4;
  STRING10  phone;
END;
```

DATASET

```
name := DATASET( file, recorddef, THOR [options]);  
name := DATASET( file, recorddef, CSV [ ( options ) ] );  
name := DATASET( file, recorddef, XML( path,[options] ) );  
name := DATASET( file, recorddef, JSON( path,[options] ) );
```

- ✓ *name* - The definition name by which the file is subsequently referenced.
- ✓ *file* - A string constant containing the logical filename.
- ✓ *recorddef* - The RECORD structure of the dataset.
- ✓ *options* - options specific to the dataset type.
- ✓ *path* - A string constant containing the full XPATH to the tag that delimits the records in the *file*
- ✓ *command* - third-party program that creates the dataset.

DATASET introduces a new data file into the system with the specified *recorddef* layout.

RECORD and DATASET example

Layout_Company := RECORD

```
UNSIGNED    sic_code;  
STRING120   company_name;  
STRING10    prim_range;  
STRING2     predir;  
STRING28    prim_name;  
STRING4     addr_suffix;  
STRING2     postdir;  
STRING5     unit_desig;  
STRING8     sec_range;  
STRING25    city;  
STRING2     state;  
STRING5     zip;  
STRING4     zip4;  
END;
```

```
EXPORT File_Company_List := DATASET('~CLASS::Company_List', Layout_Company, THOR);
```

OUTPUT

Let's display the result.

Job	Catergory	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

CHOOSSEN

Returns the first n number of records.

```
// A simple output
OUTPUT(SalaryAvg_DS, NAMED('SalaryAvg_DS'));

//CHOOSSEN
OUTPUT(CHOOSSEN(SalaryAvg_DS, 2), NAMED('SalaryAvg_Choosen'));
```

##	job	category	city	state	avg_salary
1	Manager	IT	Atlanta	GA	87000
2	Director	Art	Atlanta	GA	100000
3	CIO	IT	Tampa	FL	112000
4	Sales	General	Chicago	IL	55000

##	job	category	city	state	avg_salary
1	Manager	IT	Atlanta	GA	87000
2	Director	Art	Atlanta	GA	100000

SORT

Ascending or descending sort

Filter

Choosing a smaller part of dataset. A BOOLEAN expression following any recordset or dataset.

Job	Catergory	City	State	Avg_Salary
Manager	IT	Atlanta	GA	87000
Director	Art	Atlanta	GA	100000
CIO	IT	Tampa	FL	112000
Sales	General	Chicago	IL	55000

```
//Filter  
OUTPUT(SalaryAvg_DS(City = 'Tampa'), NAMED('Tampa_Filter'));
```

##	job	category	city	state	avg_salary
1	CIO	IT	Tampa	FL	112000

```
//Sort  
SortJobs := SORT(SalaryAvg_DS, Job);  
OUTPUT(SortJobs, NAMED('SortJobs'));
```

##	job	category	city	state	avg_salary
1	CIO	IT	Tampa	FL	112000
2	Director	Art	Atlanta	GA	100000
3	Manager	IT	Atlanta	GA	87000
4	Sales	General	Chicago	IL	55000

More on Filtering

All records within *dataset* will be evaluated

If *boolean_expression* evaluates to TRUE for a particular record, it will be included in the result

Logical Operators

AND

OR

NOT or ~

```
youngeOrLowIncome := allPeople(age < 20 OR  
| | | | | | | | | | | | avgHouseIncome <= 10000);
```

Comparison Operators

==

<> or !=

>

 \leq \geq $\langle = \rangle$

Math Functions

```
MathLayout := RECORD
```

```
  INTEGER Num1;
```

```
  INTEGER Num2;
```

```
  INTEGER Num3;
```

```
END;
```

```
DS := DATASET([ {20,45,34},  
                {909,56,45},  
                {30,-1,90} ],  
              MathLayout);
```

Num1	Num2	Num3
20	45	34
909	56	45
30	-1	90

```
COUNT(DS);           //Counts the number records in a dataset -- Returns 3  
MAX(DS, Num1);       //Returns the MAX value on a field in a dataset -- Returns 909  
MIN(DS, Num2);       //Returns the MIN value on a field in a dataset -- Returns -1  
AVE(DS, Num1);       //Returns the AVERAGE value on a field in a dataset -- Returns 319.66666666666667  
SUM(DS, Num1 + Num3); //Returns the result of adding numbers together -- Returns 1128  
TRUNCATE(AVE(DS, Num1)); //Returns the integer portion of the real_value. -- Returns 319  
ROUND(3.45);         //Returns the rounded value -- Return 3  
ROUND(3.76);         //Returns the rounded value -- Return 4
```

CORRELATION

NumOne	NumTwo
1	1
2	2
3	3
4	4
5	5
6	6



```
CORRELATION(ds1, NumOne, NumTwo)
```



Returns 1.0

NumObe	NumTwo
1938960000.00	2044820000.00
1779710000.00	854858000.00
2961810000.00	1248480000.00
2774400000.00	1263570000.00
1144160000.00	434290000.00
3387280000.00	1302380000.00
3195380000.00	1711770000.00



```
CORRELATION(ds2, NumOne, NumTwo)
```



Returns 0.4978702535543908

FUNCTION (ECL Definitions with parameters)

```
EXPORT myfunc (STRING val) := FUNCTION
| Result := 'Hello ' + val + ' , welcome to this function';
| RETURN Result;
END;

//Using myfunc
res := myfunc('Jonny');
OUTPUT(res, NAMED('res'));

OUTPUT(myfunc('Sunny'), NAMED('Sunny'));
```

<u>Sunny</u>	Hello Sunny , welcome to this function
<u>res</u>	Hello Jonny , welcome to this function

One Line Function

```
INTEGER checkMax (SET OF INTEGER numList) := MAX(numList);

OUTPUT(checkMax([2,5,8,10,45,11]), NAMED('checkMath'));
```

MODULE

Is a container that allows you to group related definitions.
The *parameters* passed to the module are shared by all the related *members* definitions.

Variable Scope

- Local definitions are visible only up to an EXPORT or SHARED
- SHARED definitions are visible within module.
- EXPORT definitions are visible within and outside of a module .

```
MyMod := MODULE

  // Visible only by MyMod
  SHARED x := 88;
  SHARED y := 42;

  // Visible by MyMod and outsiders
  EXPORT See := 'This is how a module works.';
  EXPORT res := Y * 2;
END;

OUTPUT(MyMod.See);

OUTPUT(MyMod.Res, Named('ViewResult'));
```

Result_5

This is how a module works.

ViewResult

84

TRANSFORM

Specifies exactly how each field in the output record set is to receive its value.

- It should include the result type.
- Should contain name
- Contains parameter list
- SELF: refers to fields in result type.

PROJECT

Processes through all the records in the dataset performing the TRANSFORM.

- LEFT: refers to dataset getting passed to PROJECT.
- COUNTER: Optional counter that counts calls to TRANSFORM

Standalone TRANSFORM

```
Person_Layout := RECORD
  STRING FirstName;
  STRING LastName;
END;
```

```
NameDS := DATASET([{'Sun','Shine'},
                   {'Blue','Sun'},
                   {'Silver','Rose'}],
                  Person_Layout);
```

FirstName	LastName
Sun	Shine
Blue	Moon
Silver	Rose

```
NameOutRec := RECORD
  STRING FirstName;
  STRING LastName;
  STRING CatValues;
  INTEGER RecCount
END;
```

```
NameOutRec CatThem(Person_Layout L, INTEGER C) := TRANSFORM
  SELF.CatValues := L.FirstName + ' ' + L.LastName; //Defines value for new field
  SELF.RecCount := C; // Adding Counter
  SELF := L; // Assign everything with same field name from NameDS
END;
```

```
CatRecs := PROJECT(NameDS, // Dataset to loop through
                  CatThem //Transform name
                  (LEFT, //Left dataset which is NameDS
                   COUNTER //Simpler Counter
                  ));
```

```
OUTPUT(CatRecs, NAMED('CatRecs'));
```

firstname	lastname	catvalues	reccount
Sun	Shine	Sun Shine	1
Blue	Moon	Blue Moon	2
Silver	Rose	Silver Rose	3

NameOutRec: Result Layout

CatThem: Transform Name

Person_Layout: Input Dataset Layout

L : Reference to Person_Layout fields

SELF: Refers to fields in result dataset

C: Will do the Counting

Inline TRANSFORM

```
Person_Layout := RECORD
    INTEGER PersonalID;
    STRING  FirstName;
    STRING  LastName;
END;

NameDS := DATASET([
    {100, 'Jo', 'Smith'},
    {203, 'Dan', 'Carpenter'},
    {498, 'Sally', 'Fryman'},
    {302, 'Silver', 'Rose'}],
    Person_Layout);

NameOutRec := RECORD
    INTEGER RecCount;
    INTEGER PersonalID;
    STRING  PersonName;
    STRING  FutureAddress;
END;

CatRecs := PROJECT(NameDS,
    TRANSFORM(NameOutRec,
        SELF.PersonName := LEFT.FirstName + ' ' + LEFT.LastName;
        SELF.RecCount    := COUNTER;
        SELF              := LEFT;
        SELF              := [];
    ));

OUTPUT(CatRecs, NAMED('Inline_CatRecs'));
```

PersonalID	FirstName	LastName
100	Jo	Smith
203	Dan	Carpenter
498	Sally	Fryman
302	Silver	Rose

CatRecs: Project Name

NameDS: Input Dataset to loop through

NameOutRec: Result layout

SELF: Refers to fields in result dataset

SELF := LEFT: Assign everything with same field name from NameDS

SELF := []: All un-assigned fields will be set to default values

reccount	personalid	personname	futureaddress
1	100	Jo Smith	
2	203	Dan Carpenter	
3	498	Sally Fryman	
4	302	Silver Rose	

TABLE (recordsets in memory, cross-tab tool)

```
Pickup_Layout := RECORD
  STRING10 pickup_date;
  DECIMAL8_2 fare;
  DECIMAL8_2 distance;
END;

Pickup_DS := DATASET([{'2015-01-01', 25.10, 5},
                      {'2015-01-01', 40.15, 8},
                      {'2015-01-02', 30.10, 6},
                      {'2015-01-02', 25.15, 4}],
                      Pickup_Layout);

crossTabLayout := RECORD
  Pickup_DS.pickup_date;
  avgFare := AVE(GROUP, Pickup_DS.fare);
  totalFare := SUM(GROUP, Pickup_DS.fare);
END;

crossTabDs := TABLE(Pickup_DS, // Input Dataset
                      crossTabLayout,
                      pickup_date);

OUTPUT(crossTabDs, NAMED('crossTabDs'));
```

pickup_date	fare	distance
2015-01-01	25.1	5
2015-01-01	40.15	8
2015-01-02	30.1	6
2015-01-02	25.15	4

pickup_date	avgfare	totalfare
2015-01-01	32.625	65.25
2015-01-02	27.625	55.25

JOIN

The JOIN function produces a result set based on the intersection of two or more datasets or indexes.

INNER: Only those records that exist in both datasets.

LEFT OUTER: At least one record for every record in the left.

RIGHT OUTER: At least one record for every record in the right.

LEFT ONLY: One record for each left record with no match in the left.

RIGHT ONLY: One record for each left record with no match in the right.

FULL ONLY: One record for each left and right record with no match in the opposite.

EmpDS

EmpID	Name	HireYear
1000	Jack	2014
2000	Blue	2016
3000	Mary	2016
5000	Mart	2000
8000	Cat	2002

JobCatDS

EmpID	Department	Title
1000	IT	developer
2000	Biz	Manager
4000	Fin	accountant
8000	IT	analyst

```
InnerJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT));
```

empid	name	title	department
1000	Jack	developer	IT
2000	Blue	Manager	Biz
8000	Cat	analyst	IT

```
LeftOuterJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT),
    LEFT OUTER);
```

empid	name	title	department
3000	Mary		
5000	Mart		

```
FullOuterJoin := JOIN(EmpDS, JobCatDS,
    LEFT.EmpID = RIGHT.EmpID,
    TRANSFORM(EmpResult_Layout,
        SELF := LEFT,
        SELF := RIGHT),
    FULL OUTER);
```

empid	name	title	department
1000	Jack	developer	IT
2000	Blue	Manager	Biz
3000	Mary		
0		accountant	Fin
5000	Mart		
8000	Cat	analyst	IT

VISUALIZATION (built-ins and an ECL Bundle)

Methods include

- Two-Dimensional
- Multi-Dimensional Methods
- Geospatial
- General

A basic visualization typically requires the following steps:

1. Creation of a suitable dataset.
2. Output the dataset with a suitable name, so that visualization can locate the data.
3. Create (and output) the visualization, referencing the named output from step 2

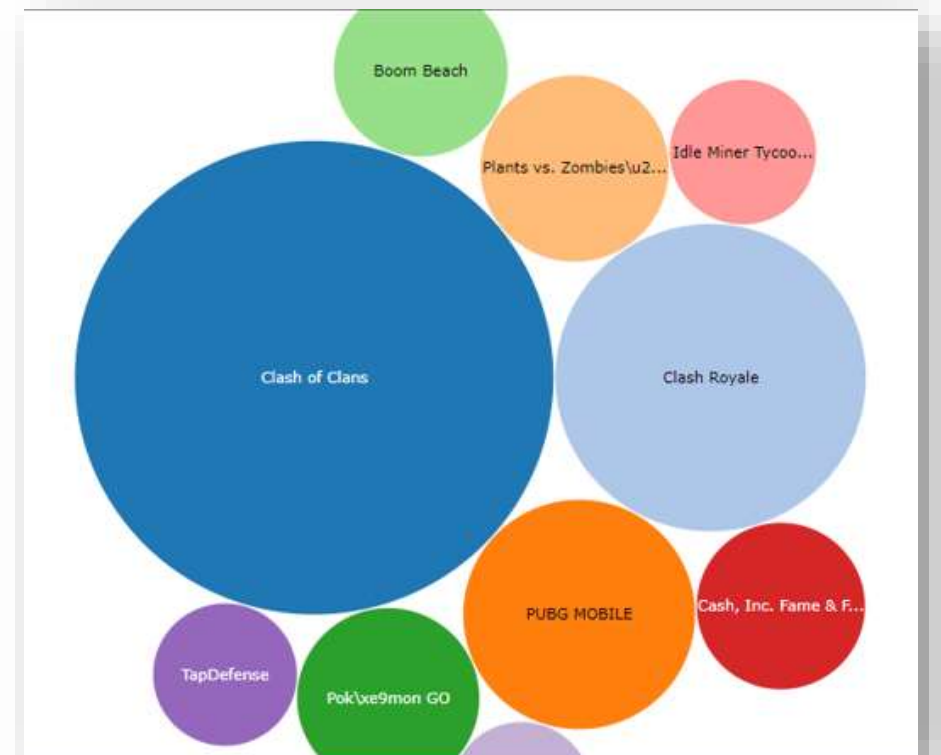
```

top_user_rating_count := TOPN(
    TABLE(clean_mod.games_ds,
        {name,
         user_rating_count}) ,
    10,
    -user_rating_count);

OUTPUT(analysis_mod.top_user_rating_count, NAMED('user_rating_count'));
Visualizer.TwoD.Bubble(['user_rating_count',
    /*datasource*/,
    'user_rating_count']);

```

Bubble
 Pie
 Bar
 Scatter
 Line
 WorldCloud
 Area





Safe Haven Challenge

FEBRUARY 2024

Bob Foreman
Software Engineer Lead
LexisNexis Risk Solutions

Safe Havens!

Many travelers find themselves on occasion in a strange land and a strange city, sometimes without their control. Without knowing the risks and dangers in that area, sometimes a tourist can suddenly be in a dangerous situation.

What can we do as developers to help prevent this?

This year's challenge will analyze different social factors by area such as poverty, unemployment, and many other factors to assess the risk for the traveler in a strange land.

The goal of the challenge is to answer two questions:

1. Analysis of social factors in an area (unemployment, education, poverty, and population) and identify it as a "Hot Spot".
2. Provide additional information to the traveler to help find “safe haven” resources in their area (fire and police stations, hospitals, churches, food banks, etc.)

The Data!

City and County Data has been collected from all 50 US states and organized into a simple dataset to use as your source. In addition, many public datasets have also been gathered and cleaned to help get you started.

These datasets include:

Education

Unemployment

Poverty

Population

Crime

Police

Fire Stations

Hospitals

Places of Worship

Food Banks

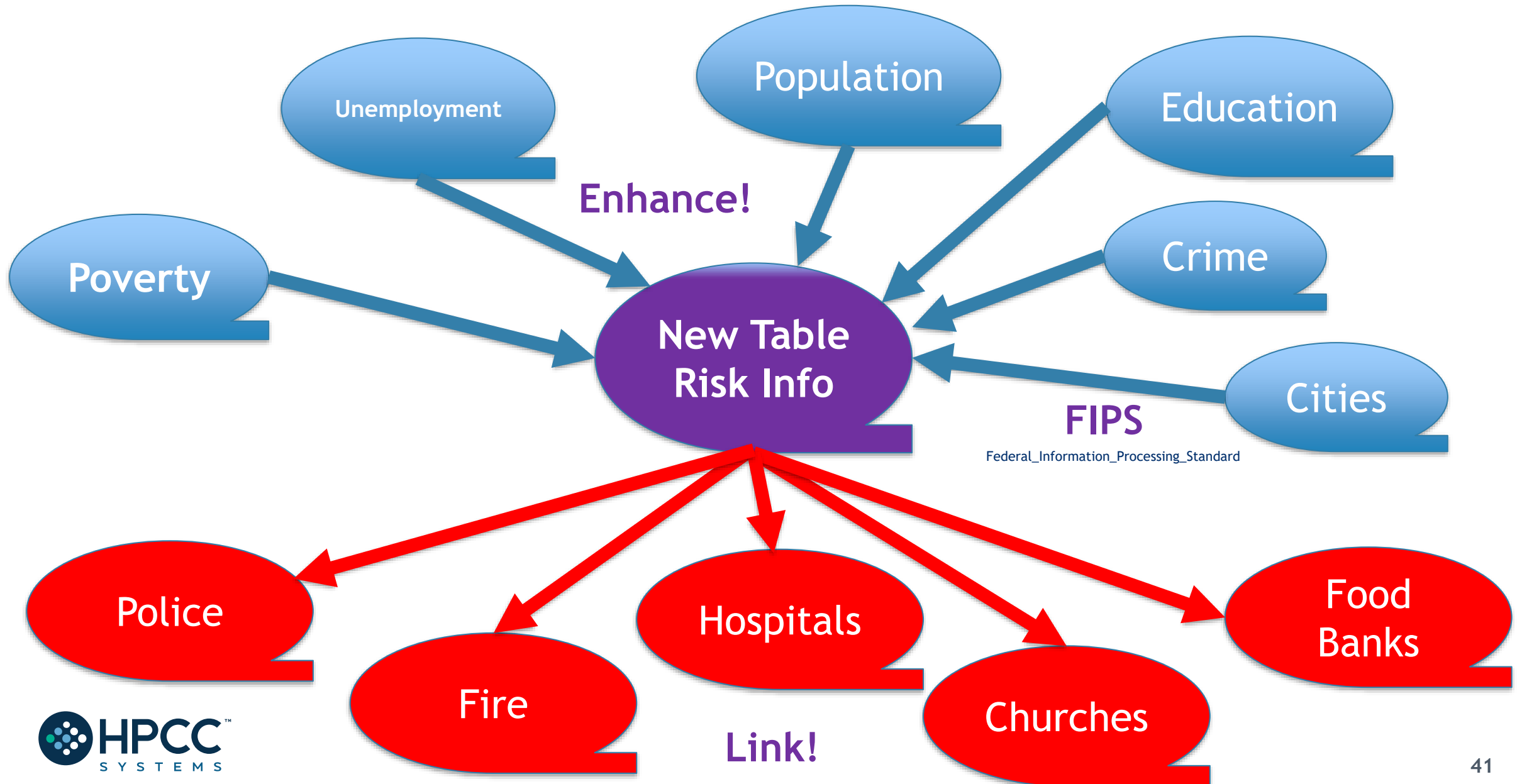
Reference Datasets:

A **Cities** dataset with related FIPS and Zip Codes (used for linking the above datasets to a new “Risk” dataset)

Unemployment Rates (Not really used in this challenge but interesting data!)

You are not limited to using these datasets! Extra credit will be rewarded by linking in other pertinent data!

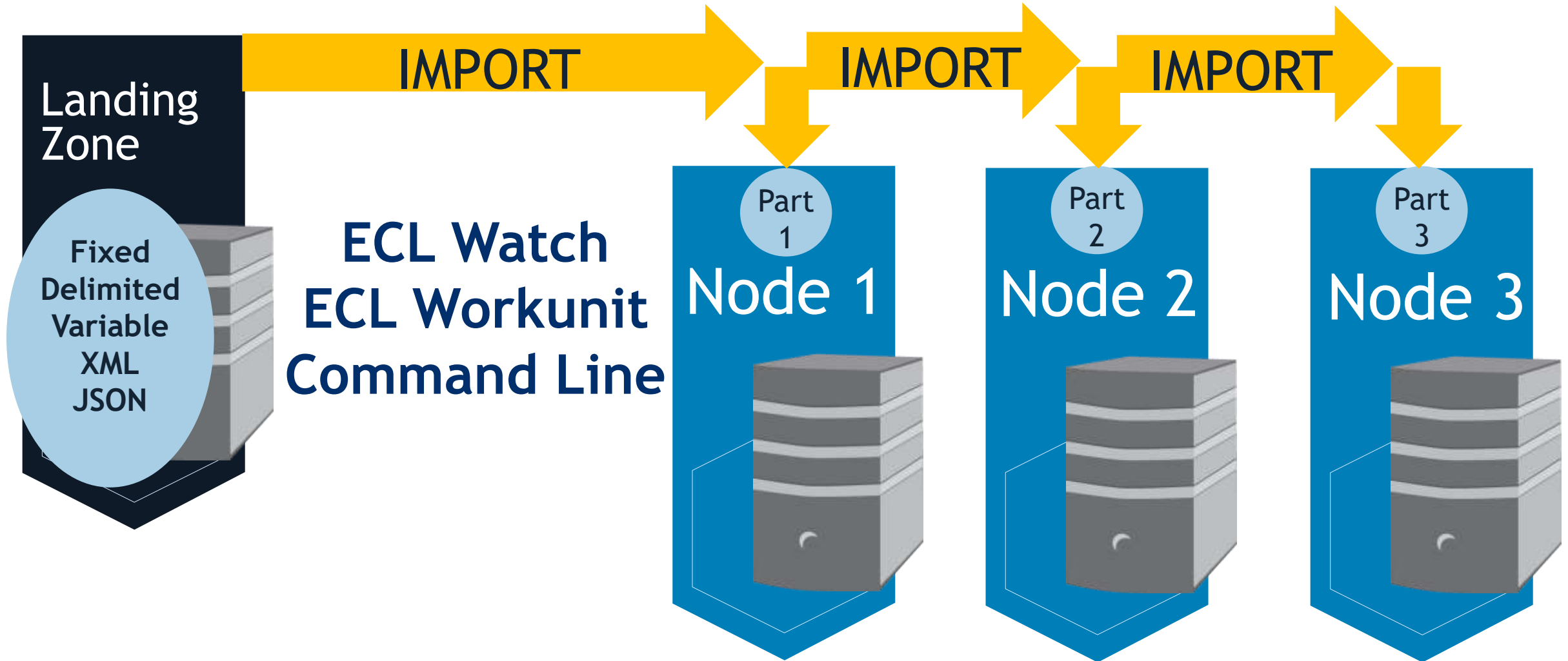
Data Flow





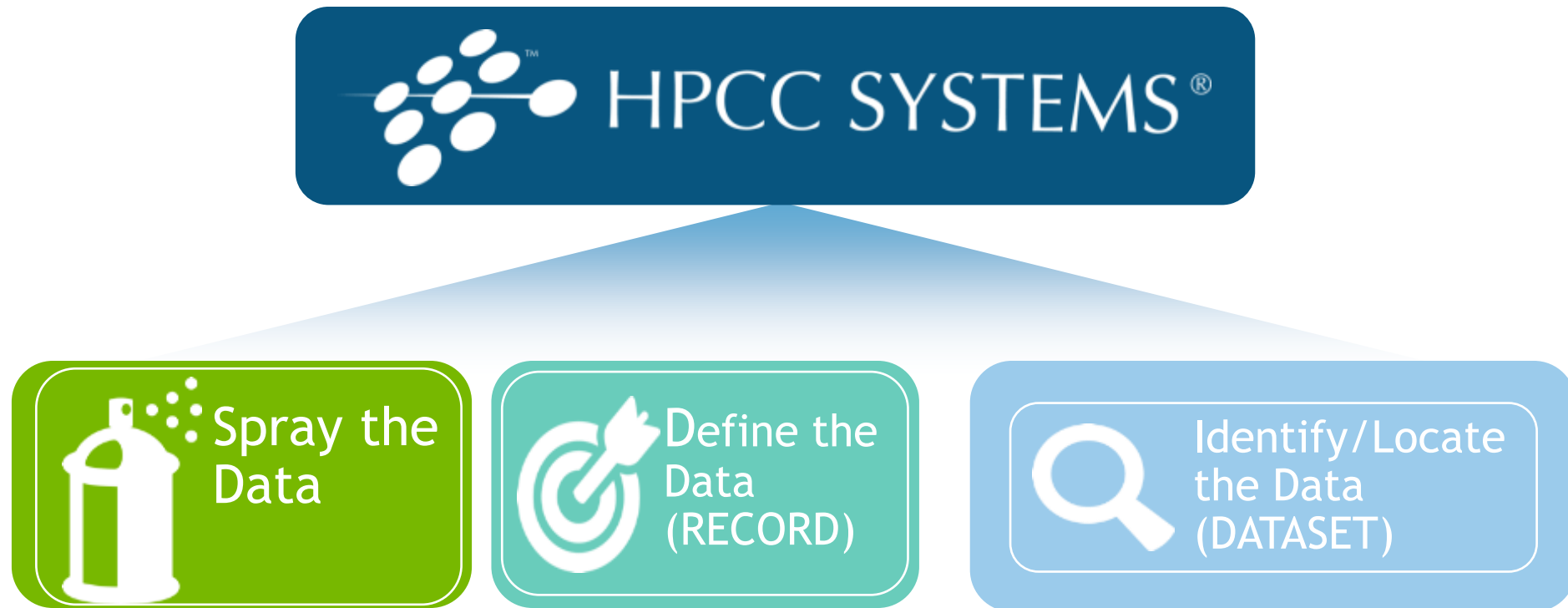
Creating the Project

Getting the data to the cluster!



Three ECL Data Rules

Before you begin to work on any data in the HPCC cluster, you must always do three things:



Phase I: Import and Define the Data

File_AllData defines all input sources.

```
25 //https://www.ers.usda.gov/data-products/county-level-data-sets/county-level-data-sets-download-data/
26 //Unemployment stats from 2000-2021
27 □EXPORT unemp_byCounty := RECORD
35   EXPORT unemp_byCountyDS := DATASET('~safe::in::unemployment',unemp_byCounty,CSV(HEADING(1)));
36
37 □EXPORT pov_estimates := RECORD
45   EXPORT pov_estimatesDS := DATASET('~safe::in::poverty',pov_estimates,CSV(HEADING(1)));
46
47 □EXPORT Education := RECORD
55   EXPORT EducationDS := DATASET('~safe::in::education',education,CSV(HEADING(1)));
56
57 □EXPORT pop_estimates := RECORD
58   UNSIGNED3 FIPS_Code;
59   STRING2   State;
60   STRING50  Area_Name;
61   STRING35  Attribute;
62   REAL8     Value;
63 END;
64
65   EXPORT pop_estimatesDS := DATASET('~safe::in::population',pop_estimates,CSV(HEADING(1)));
66
67 //https://hifld-geoplatform.hub.arcgis.com/datasets
```

I

Viewing the Data

BWR_AllInputData views all input sources:

```
1  IMPORT $;
2  SAFE := $.File_AllData; //See this file for you data dictionary
3  //RISK:
4  OUTPUT(SAFE.unemp_byCountyDS,NAMED('Unemployment'));
5  OUTPUT(SAFE.EducationDS,NAMED('Education'));
6  OUTPUT(SAFE.pov_estimatesDS,NAMED('Poverty'));
7  OUTPUT(SAFE.pop_estimatesDS,NAMED('Population'));
8  OUTPUT(SAFE.CrimeDS,NAMED('Crime'));
9  //RESOURCES:
10 OUTPUT(SAFE.PoliceDS,NAMED('Police'));
11 OUTPUT(SAFE.FireDS,NAMED('Fire'));
12 OUTPUT(SAFE.HospitalDS,NAMED('Hospitals'));
13 OUTPUT(SAFE.ChurchDS,NAMED('Churches'));
14 OUTPUT(SAFE.FoodBankDS,NAMED('FoodBanks'));
15 //REFERENCE:
16 OUTPUT(SAFE.City_DS,NAMED('Cities'));
17 OUTPUT(SORT(SAFE.City_DS,county_fips),NAMED('FipsCities'));
18 OUTPUT(COUNT(SAFE.City_DS),NAMED('Cities_Cnt'));
19
```

Phase II: Analyze and Build the Risk Data

BWR_Analyze_Education

BWR_Analyze_Population

BWR_Analyze_Poverty_Data

BWR_Analyze_Population

BWR_CreateCoreFile

Phase III: Linking the Data

Building the “core” data:

BWR_CreateCoreFile

Scoring and Adjustments

```
22 RiskPlusRec := RECORD
23   BaseInfo;
24   EducationScore := 0;
25   PovertyScore   := 0;
26   PopulationScore := 0;
27   CrimeScore     := 0;
28   Total          := 0;
29 END;
30
31 RiskTbl := TABLE(BaseInfo,RiskPlusRec);
32 OUTPUT(RiskTbl,NAMED('BuildTable'));
33
34 //Let's add a Crime Score!
35
36 CrimeRec := RECORD
37   CrimeRate := TRUNCATE((INTEGER)Crime.crime_rate_per_100000);
38   Crime.fips_st;
39   fips_cty := (INTEGER)Crime.fips_cty;
40   Fips := Crime.fips_st + INTFORMAT(Crime.fips_cty,3,1);
41 END;
42
43 CrimeTbl := TABLE(Crime,CrimeRec);
44 OUTPUT(CrimeTbl,NAMED('BuildCrimeTable'));
45
46 JoinCrime := JOIN(CrimeTbl,RiskTbl,
47   LEFT.fips = (STRING5)RIGHT.county_fips,
48   TRANSFORM(RiskPlusRec,
49     SELF.CrimeScore := LEFT.crimerate,
50     SELF              := RIGHT),
51   RIGHT OUTER);
52
53 OUTPUT(SORT(JoinCrime,-CrimeScore),NAMED('AddedCrimeScore'));
```


Phase III: Analyzing and Scoring the Risk

Create a Core Dataset that aggregates your analysis:

```
1  // Let's create a core "risk" file that the county code (FIPS) and the primary city.
2  // We can extra ct this data from the Cities file.
3  IMPORT $;
4  CityDS := $.File_AllData.City_DS;
5  Crime  := $.File_AllData.CrimeDS;
6
7  //CityDS(county_fips = 5035); Test to verify data accuracy for the crime score
8
9
10 // Declare our core RECORD:
11 RiskRec := RECORD
12     STRING45 city;
13     STRING2  state_id;
14     STRING20 state_name;
15     UNSIGNED3 county_fips;
16     STRING30 county_name;
17 END;
18
19 BaseInfo := PROJECT(CityDS,RiskRec);
20 OUTPUT(BaseInfo,NAMED('BaseData'));
21
22 RiskPlusRec := RECORD
23     BaseInfo;
24     EducationScore := 0;
25     PovertyScore   := 0;
26     PopulationScore := 0;
27     CrimeScore     := 0;
28     Total          := 0;
29 END;
30
31 RiskTbl := TABLE(BaseInfo,RiskPlusRec);
32 OUTPUT(RiskTbl,NAMED('BuildTable'));
```

Phase IV: Prepare the Havens

```
10  EXPORT File_CleanHavens := MODULE
11  //This module is used to clean the File, Police, and Hospital datasets to be indexed and used in the NCMEC ROXIE service
12  //DATASETS generated in BWR_GenerateCleanResponders, but declared here.
13  //INDEXes built in BWR_BuildIndexes, but also declared here.
14  SHARED CleanFireRec := RECORD
15      STRING100 name;
16      STRING60  addressbuildingname;
17      STRING65  address;
18      UNSIGNED3 PrimaryFIPS := 0; //New - Added from Cities DS
19      STRING35  city;
20      STRING2   state;
21      STRING10  zipcode;
22  END;
23  EXPORT CleanFire := PROJECT(Fire, TRANSFORM(CleanFireRec,
24      SELF.name           := STD.STR.ToUpperCase(LEFT.name),
25      SELF.addressbuildingname := STD.STR.ToUpperCase(LEFT.addressbuildingname),
26      SELF.address        := STD.STR.ToUpperCase(LEFT.address),
27      SELF.city           := STD.STR.ToUpperCase(LEFT.city),
28      SELF.State          := STD.STR.ToUpperCase(LEFT.state),
29      SELF.zipcode        := STD.STR.ToUpperCase(LEFT.zipcode)));
30  EXPORT CleanFireFIPS := JOIN(CleanFire, Cities,
31      LEFT.city = STD.STR.ToUpperCase(RIGHT.city) AND
32      LEFT.state = RIGHT.state_id,
33      TRANSFORM(CleanFireRec,
34          SELF.PrimaryFIPS := (UNSIGNED3)RIGHT.county_fips,
35          SELF             := LEFT), LEFT OUTER, LOOKUP);
36
37  EXPORT CleanFireDS := DATASET('~SAFE::OUT::Fire', CleanFireRec, FLAT);
```

Phase IV: Prepare the Havens

BWR_GenerateCleanHavens

```
1  IMPORT $;  
2  
3  OUTPUT($.File_CleanHavens.CleanFireFIPS,, '~SAFE::OUT::Fire', OVERWRITE);  
4  OUTPUT($.File_CleanHavens.CleanPolice,, '~SAFE::OUT::Police', OVERWRITE);  
5  OUTPUT($.File_CleanHavens.CleanHospital,, '~SAFE::OUT::Hospital', OVERWRITE);  
6  OUTPUT($.File_CleanHavens.CleanChurchFIPS,, '~SAFE::OUT::Churches', OVERWRITE); //Churches  
7  // OUTPUT() //Foodbanks  
8  
9  //NOTE: DATASETs and INDEXes defined in File_CleanHavens MODULE.  
10
```




Data Delivery (Roxie and Visualization)

Delivering your Results!

Phase V: - Declare and Build your Indexes (Churches):

```
97 //Churches
98 SHARED CleanChurchRec := RECORD
99     STRING70 name;
100     STRING35 street;
101     STRING22 city;
102     STRING2 state;
103     UNSIGNED3 zip;
104     UNSIGNED1 affiliation;
105     UNSIGNED3 PrimaryFIPS; //New - will be added from Cities DS
106 END;
107 //PROJECT is used to transform one data record to another.
108 CleanChurch := PROJECT Churches, TRANSFORM(CleanChurchRec,
109     SELF.name := STD.STR.ToUpperCase(LEFT.name),
110     SELF.street := STD.STR.ToUpperCase(LEFT.street),
111     SELF.city := STD.STR.ToUpperCase(LEFT.city),
112     SELF.State := STD.STR.ToUpperCase(LEFT.state),
113     SELF.zip := LEFT.zip,
114     SELF.affiliation := LEFT.affiliation,
115     SELF.PrimaryFIPS := 0));
116 //JOIN is used to combine data from different datasets
117 EXPORT CleanChurchFIPS := JOIN(CleanChurch, Cities,
118     LEFT.city = STD.STR.ToUpperCase(RIGHT.city) AND
119     LEFT.state = RIGHT.state_id,
120     TRANSFORM(CleanChurchRec,
121         SELF.PrimaryFIPS := (UNSIGNED3)RIGHT.county_fips,
122         SELF := LEFT), LEFT OUTER, LOOKUP);
123 EXPORT CleanChurchesDS := DATASET('~SAFE::OUT::Churches', CleanChurchRec, FLAT);
124
125 //Declare and Build Indexes (special datasets that can be used in the ROXIE data delivery cluster)
126 EXPORT CleanChurchIDX := INDEX(CleanChurchesDS, {city, state}, {CleanChurchesDS}, '~SAFE::IDX::Church::CityPay');
127 EXPORT CleanChurchFIPSIDX := INDEX(CleanChurchesDS, {PrimaryFIPS}, {CleanChurchesDS}, '~SAFE::IDX::Church::FIPSPay');
128 EXPORT BuildChurchIDX := BUILD(CleanChurchIDX, OVERWRITE);
129 EXPORT BuildChurchFIPSIDX := BUILD(CleanChurchFIPSIDX, OVERWRITE);
```

Delivering your Results!

Phase V: Declare and Build your Indexes:

```
1  IMPORT $;
2  //Indexes used for ROXIE demo
3
4  $.File_CleanResponders.BuildFireIDX;
5  $.File_CleanResponders.BuildFireFIPSIDX;
6  $.File_CleanResponders.BuildPoliceIDX;
7  $.File_CleanResponders.BuildHospitalIDX;
8  $.File_CleanResponders.BuildChurchIDX;
9  $.File_CleanResponders.BuildChurchFIPSIDX;
10
```

##	countyfips	city	state	name	address	zip	zip4	telephone	type	status	population	county	country	latitude
1	1001	PRATTVILLE	AL	PRATTVILLE BAPTIST HOSPITAL	124 SOUTH MEMORIAL DRIVE	36067	NOT AVAILABLE	(334) 361-4267	GENERAL ACUTE CARE	OPEN	107	AUTAUGA	USA	32.4638
2	1003	BAY MINETTE	AL	NORTH BALDWIN INFIRMARY	1815 HAND AVENUE, PO BOX 1409	36507	NOT AVAILABLE	(251) 937-5521	GENERAL ACUTE CARE	OPEN	78	BALDWIN	USA	30.9839
3	1003	DAPHNE	AL	EASTPOINTE HOSPITAL	7400 ROPER LANE	36526	NOT AVAILABLE	(251) 450-5901	PSYCHIATRIC	OPEN	66	BALDWIN	USA	30.6352
4	1003	FAIRHOPE	AL	THOMAS HOSPITAL	750 MORPHY AVENUE, PO DRAWER 929	36532	NOT AVAILABLE	(251) 279-1501	GENERAL ACUTE CARE	OPEN	176	BALDWIN	USA	30.5188
5	1003	FOLEY	AL	SOUTH BALDWIN REGIONAL MEDICAL CENTER	1613 NORTH MCKENZIE STREET	36535	NOT AVAILABLE	(251) 949-3400	GENERAL ACUTE CARE	OPEN	112	BALDWIN	USA	30.4258
6	1005	EUFULA	AL	MEDICAL CENTER BARBOUR	820 WEST WASHINGTON STREET	36027	NOT AVAILABLE	(334) 688-7132	GENERAL ACUTE CARE	OPEN	74	BARBOUR	USA	31.8886
7	1007	CENTREVILLE	AL	BIBB MEDICAL CENTER	208 PIERSON AVENUE	35042	NOT AVAILABLE	(205) 926-4881	GENERAL ACUTE CARE	OPEN	35	BIBB	USA	32.9513
8	1009	ONEONTA	AL	ST. VINCENT'S BLOUNT	150 GILBREATH DRIVE, PO BOX 1000	35121	NOT AVAILABLE	(205) 274-3000	CRITICAL ACCESS	OPEN	40	BLOUNT	USA	33.9296
9	1011	UNION SPRINGS	AL	BULLOCK COUNTY HOSPITAL	102 WEST CONECUH AVENUE	36089	NOT AVAILABLE	(334) 738-2140	GENERAL ACUTE CARE	OPEN	61	BULLOCK	USA	32.1466

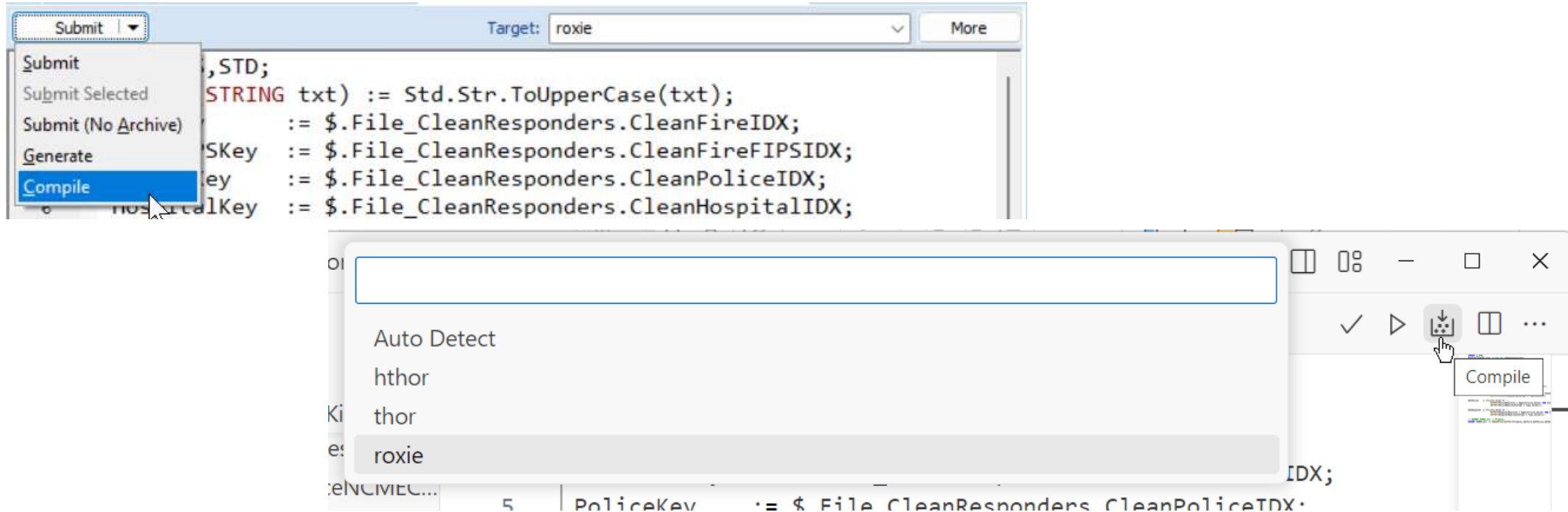
Delivering your Results!

Phase VI: - Design and Write Your Query:

```
1 //TO DO - Add Food Banks
2 IMPORT $,STD;
3 UpperIt(STRING txt) := Std.Str.ToUpperCase(txt);
4 CoreFIPSTKey := $.File_SvcData.CleanCoreFIPSTKey;
5 CoreCitySTKey := $.File_SvcData.CleanCitySTKey;
6 FireKey := $.File_CleanResponders.CleanFireKey;
7 FireFIPSTKey := $.File_CleanResponders.CleanFireFIPSTKey;
8 PoliceKey := $.File_CleanResponders.CleanPoliceKey;
9 HospitalKey := $.File_CleanResponders.CleanHospitalKey;
10 ChurchCitySTKey := $.File_CleanResponders.CleanChurchKey;
11 ChurchFIPSTKey := $.File_CleanResponders.CleanChurchFIPSTKey;
12
13 UNSIGNED3 fips_value := 0 : STORED('FIPS');
14 STRING23 city_value := '' : STORED('City');
15 STRING2 state_value := '' : STORED('State');
16
17 GetPrimary := IF(fips_value = 0,
18     OUTPUT(CoreCitySTKey(City = UpperIt(city_value) AND State_id = UpperIt(state_value)),NAMED('CoreDataByCity')),
19     OUTPUT(CoreFIPSTKey(county_fips = fips_value),NAMED('CoreDataByFIPS')));
20
21 GetFire := IF(city_value <> '',
22     OUTPUT(FireKey(City = UpperIt(city_value) AND State = UpperIt(state_value)),NAMED('Fire_Stations_ByCity')),
23     OUTPUT(FireFIPSTKey(primary_fips = fips_value),NAMED('Fire_Stations_ByFIPS')));
24
25 GetPolice := IF(city_value <> '',
26     OUTPUT(PoliceKey(City = UpperIt(city_value) AND State = UpperIt(state_value) AND WILD(county_fips)),NAMED('Police_Stations_ByCity')),
27     OUTPUT(PoliceKey(county_fips = fips_value),NAMED('Police_Stations_ByFIPS')));
28
29 GetHospital := IF(city_value <> '',
30     OUTPUT(HospitalKey(City = UpperIt(city_value) AND State = UpperIt(state_value) AND WILD(county_fips)),NAMED('Hospitals_ByCity')),
31     OUTPUT(HospitalKey(county_fips = fips_value),NAMED('Hospitals_ByFIPS')));
32
33 GetChurches := IF(Fips_Value = 0,
34     OUTPUT(ChurchCitySTKey(City=UpperIt(City_value),State=UpperIt(State_value)),NAMED('Worship_ByCity')),
35     OUTPUT(ChurchFIPSTKey(PrimaryFIPS=Fips_Value),NAMED('Worship_ByFIPS')));
36
37 EXPORT SafeHaven_Svc := SEQUENTIAL(GetPrimary,GetFire,GetPolice,GetHospital,GetChurches);
38
```

Delivering your Results!

Phase VII: - Deploy(Publish) and then Test Your Query:



Delivering your Results!

Phase VII: - Deploy(Publish) and then Test Your Query:

ie | Publish | Z.A.P | Worker Logs

Job Name: DEMO.Safe_Svc

Remote Dali:

Source Process:

Comment:

Priority: None

Allow Foreign Files: ☒

Update Super Files: ☐

Submit

Delivering your Results!

Phase VII - Deploy(Publish) and then Test Your Query:

The screenshot shows the HPCC Systems web interface in a browser. The address bar displays the URL `training.us-hpccsystems-dev.azure.lnrsg.io:8002`, with a red arrow pointing to it. The browser's bookmark bar includes 'Corporate Bookmarks', 'System Dashboard...', 'Home Page | HPCC...', 'ECL Watch Training 2', 'ECL Watch | 160', and 'Eventbrite'. The HPCC Systems header includes 'View', 'Frame', and 'Log Out' links. Below the header, there are tabs for 'Form' and 'Links'. The left sidebar, titled 'Active Queries', shows a tree structure under 'Targets' with 'hthor' expanded, containing 'demo.safe_svc' (highlighted with a red arrow), 'thor', 'roxie', and 'thor_roxie'. The main content area displays the 'hthor' target configuration, showing 'demo.safe_svc' with a 'Dynamic Form' dropdown. Below this, the 'DEMO_SAFE_SVCREQUEST' section contains three input fields: 'cityval:', 'fipsval:', and 'stateval:'. A red arrow points to the 'Submit' button at the bottom of the form. The bottom of the interface includes 'Output Tables' and 'FORM POST' dropdowns, and 'Submit' and 'Clear All' buttons.

Phase VII - Deploy(Publish) and then Test Your Query:

oedec.nsfchaven inc.1 Response

Dataset: CoreDataByCity

city	state id	state name	county fips	county name	education score	poverty score	crime score	trial score	income fips
BOSTON	MA	MASSACHUSETTS	25025	SUFFOLK	15.3	38.05	46.4	37.67	0

Dataset: Fire_Stations_By_City

[illegible]

Dataset: Police_Stations_By_City

id	agency	city	state	name	address	zip	zip3	zip4	zipcode	type	area	county	county	city	lat	lon	status
1	25000	BOSTON	MA	ARTHUR BULFORD STATION	2700 MASSACHUSETTS AVENUE	02114	2388		02114-2388	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.350333	-71.069111	0
2	25000	BOSTON	MA	BOSTON POLICE - INBOARD PATROL	30 DEVONPORT AVENUE	02128	2393		02128-2393	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.375717	-71.038600	0
3	25000	BOSTON	MA	BOSTON POLICE DEPARTMENT - HEADQUARTERS	1 SCHOFIELD PLAZA	02130	2414		02130-2414	LOCAL POLICE DEPARTMENT	2863	STUTFORD	STUTFORD	BOSTON	42.352638	-71.062688	0
4	25000	BOSTON	MA	BOSTON POLICE DEPARTMENT - HEADQUARTERS-T-1	45 NEW SUBURY STREET	02114	2812		02114-2812	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.341917	-71.057589	0
5	25000	BOSTON	MA	BOSTON POLICE DEPARTMENT - HEADQUARTERS-T-2	608 HARRISON AVENUE	02118	2834		02118-2834	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.347817	-71.057183	0
6	25000	BOSTON	MA	BOSTON SCHOOL POLICE	26 COURT STREET	02108	2848	0000	02108-2848	SPECIAL JURISDICTION	84	STUTFORD	STUTFORD	BOSTON	42.358794	-71.068933	0
7	25000	BOSTON	MA	BOSTON UNIVERSITY CAMPUS POLICE	32 HARRY AGASSIS WAY	02115	1368		02115-1368	SPECIAL JURISDICTION	4600	STUTFORD	STUTFORD	BOSTON	42.352516	-71.064126	0
8	25000	BOSTON	MA	BUNIER HILL COMMUNITY COLLEGE CAMPUS POLICE	205 NEW BETHFORD AVENUE	02134	2841	0000	02134-2841	SPECIAL JURISDICTION	28	STUTFORD	STUTFORD	BOSTON	42.374578	-71.088544	0
9	25000	BOSTON	MA	BUSINESS COLLEGE CAMPUS POLICE	100 BEACON STREET	02108	2848	0000	02108-2848	SPECIAL JURISDICTION	4600	STUTFORD	STUTFORD	BOSTON	42.358794	-71.068933	0
10	25000	BOSTON	MA	MASSACHUSETTS BAY TRANSPORTATION AUTHORITY TRANSIT POLICE DEPARTMENT	200 SOUTH STATION AVENUE	02110	2733		02110-2733	SPECIAL JURISDICTION	305	STUTFORD	STUTFORD	BOSTON	42.319251	-71.064507	0
11	25000	BOSTON	MA	MASSACHUSETTS COLLEGE OF ARTS CAMPUS POLICE - PUBLIC SAFETY	621 HUNTINGTON AVENUE	02115	5811		02115-5811	SPECIAL JURISDICTION	22	STUTFORD	STUTFORD	BOSTON	42.358845	-71.068408	0
12	25000	BOSTON	MA	MASSACHUSETTS ENVIRONMENTAL POLICE - HEADQUARTERS	25 CALDWELL STREET SUITE 105	02114	2819		02114-2819	SPECIAL JURISDICTION	021	STUTFORD	STUTFORD	BOSTON	42.363100	-71.058001	0
13	25000	BOSTON	MA	MASSACHUSETTS ENVIRONMENTAL POLICE - PUBLIC SAFETY	100 BLACK FRIAR STREET	02114	2819		02114-2819	SPECIAL JURISDICTION	021	STUTFORD	STUTFORD	BOSTON	42.363100	-71.058001	0
14	25000	BOSTON	MA	MASSACHUSETTS PORT AUTHORITY POLICE - BOSTON AIRPORT	101 TERRILL STREET	02114	1884		02114-1884	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.363819	-71.064507	0
15	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - BOSTON	208 LEVERTH CIRCLE	02114	1880		02114-1880	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.363819	-71.064507	0
16	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - BOSTON	1848 SOLDIER FIELD AVENUE	02115	1827		02115-1827	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.363819	-71.064507	0
17	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - BOSTON	145 BOSTON AVENUE	02114	1884		02114-1884	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.363819	-71.064507	0
18	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - LOGAN	485 TERRILL CIRCLE	02114	1880		02114-1880	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.363819	-71.064507	0
19	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - SOUTH BOSTON	125 MILLIS JAY BOULEVARD	02135	2113		02135-2113	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.322158	-71.087838	0
20	25000	BOSTON	MA	MASSACHUSETTS STATE POLICE - SOUTH BOSTON	189 WASHINGTON ROAD	02135	2652		02135-2652	PRIMARY STATE AGENCY	4600	STUTFORD	STUTFORD	BOSTON	42.346803	-71.036675	0
21	25000	BOSTON	MA	NORFOLK COUNTY INVESTIGATIVE DEPARTMENT	734 COLUMBIA AVENUE	01901	2814		01901-2814	SPECIAL JURISDICTION	4600	STUTFORD	STUTFORD	BOSTON	42.347817	-71.057183	0
22	25000	BOSTON	MA	SIMMONS COLLEGE DEPARTMENT OF PUBLIC SAFETY	1 PEARCE ROAD	02115	2620		02115-2620	SPECIAL JURISDICTION	4600	STUTFORD	STUTFORD	BOSTON	42.338712	-71.087138	0
23	25000	BOSTON	MA	SUFFOLK COUNTY HOUSE OF CORRECTION	38 BRANTON STREET	01818	2535		01818-2535	SHERIFF'S OFFICE	1054	STUTFORD	STUTFORD	BOSTON	43.134170	-70.800201	0
24	25000	BOSTON	MA	SUFFOLK COUNTY SUPERIOR COURT	130 BRANTON SQUARE	01818	2535		01818-2535	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	43.130941	-71.078123	0
25	25000	BOSTON	MA	SUFFOLK COUNTY SUPERIOR COURT - JUDICIAL DEPARTMENT	130 BRANTON SQUARE	01818	2535		01818-2535	SPECIAL JURISDICTION	021	STUTFORD	STUTFORD	BOSTON	43.130941	-71.078123	0
26	25000	BOSTON	MA	THE TUFTS UNIVERSITY POLICE DEPARTMENT	138 HARRISON AVENUE	02111	1817		02111-1817	LOCAL POLICE DEPARTMENT	4600	STUTFORD	STUTFORD	BOSTON	42.359660	-71.061425	0

Dataset: Hospitals_By_City

agency	city	name	address	zip	zip4	mapname	general	alt	intest	position	county	country	latitude	longitude	status
1	ROSTON	BETH ISRAEL DEACONESSIAN CRT- WEST	101 BETH ISRAEL DEACONESSIAN CRT- WEST	02128	02128	02128	GENERAL	ALT	0		MA	USA	42.3379	-71.1582	0
2	ROSTON	BETH ISRAEL DEACONESSIAN CRT- EAST	101 BETH ISRAEL DEACONESSIAN CRT- EAST	02128	02128	02128	GENERAL	ALT	0		MA	USA	42.3379	-71.1582	0
3	ROSTON	BOSTON CHILDREN'S HOSPITAL	300 LONGWOOD AVENUE	02115	02115	02115	CHILDREN		0		MA	USA	42.3372	-71.06388888888889	0
4	ROSTON	BOSTON MICH CTR CORP MEDICAL RESEARCH	350 HARRISON AVENUE	02118	02118	02118	GENERAL	ALT	0		MA	USA	42.3347	-71.15222222222222	0
5	ROSTON	BOSTON MICH CTR CORP MEDICAL RESEARCH	350 HARRISON AVENUE	02118	02118	02118	GENERAL	ALT	0		MA	USA	42.3347	-71.15222222222222	0
6	ROSTON	BRIGHTMAN AND WOMEN'S SKINNER HOSPITAL	115 CENTRE STREET	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.315	-71.1283	0
7	ROSTON	BRIGHTMAN AND WOMEN'S SKINNER HOSPITAL	75 FRANCIS STREET	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.3358	-71.15888888888889	0
8	ROSTON	BRIGHTMAN AND WOMEN'S SKINNER HOSPITAL	115 CENTRE STREET	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.315	-71.1283	0
9	ROSTON	DANA-FARBER CANCER INSTITUTE	450 BROOKLINE AVENUE	02115	02115	02115	GENERAL	ALT	0		MA	USA	42.33715	-71.07816	0
10	ROSTON	DR SOLIMAN CAPTER FULLER MEDICAL HEALTH CENTER	85 EAST NEWTON STREET	02459	02459	02459	PSYCHIATRIC		0		MA	USA	42.336271	-71.18163333333333	0
11	ROSTON	FRANCISCAN HEALTHCARE CENTER	200 CENTRE STREET	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.3358	-71.15888888888889	0
12	ROSTON	HARBEN REHABILITATION CENTER	1300 CENTRE STREET	02130	02130	02130	REHABILITATION		0		MA	USA	42.3665055	-71.17733333333333	0
13	ROSTON	KINDRED HOSPITAL, BOSTON	1515 COMMONWEALTH AVENUE	02130	02130	02130	LONG TERM CARE	CLOUSED	5		MA	USA	42.3366	-71.16100000000001	0
14	ROSTON	MASSACHUSETTS GENERAL HOSPITAL	1300 CENTRE STREET	02130	02130	02130	PSYCHIATRIC		0		MA	USA	42.3360	-71.17177777777778	0
15	ROSTON	MASAGUASHO'S EYE AND EAR GENERAL - MARK CAMP	343 CHALICE STREET	02114	02114	02114	GENERAL	ALT	0		MA	USA	42.3333	-71.16666666666667	0
16	ROSTON	MASAGUASHO'S EYE AND EAR GENERAL - MARK CAMP	351 BRUIT STREET	02114	02114	02114	GENERAL	ALT	0		MA	USA	42.3333	-71.16666666666667	0
17	ROSTON	NEW ENGLAND PARTHARTER HOSPITAL	125 PARKER HILL AVENUE	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.3297	-71.17016666666667	0
18	ROSTON	NEW ENGLAND PARTHARTER HOSPITAL	300 WEST AVENUE	02130	02130	02130	GENERAL	ALT	0		MA	USA	42.3297	-71.17016666666667	0
19	ROSTON	ST JULIAN'S MEDICAL CENTER	716 CAMBRIDGE STREET	02114	02114	02114	GENERAL	ALT	0		MA	USA	42.3387	-71.1488	0
20	ROSTON	THE ARMOIR HOSPITAL	616 BROADWOOD AVENUE	02130	02130	02130	PSYCHIATRIC		0		MA	USA	42.318181	-71.172311	0
21	ROSTON	THE BOSTON CENTER FOR SENIOR CITIZENS	800 WASHINGTON STREET	02114	02114	02114	GENERAL	ALT	0		MA	USA	42.3399	-71.06388888888889	0
22	ROSTON	TUFTS MEDICAL CENTER	800 WASHINGTON STREET	02114	02114	02114	GENERAL	ALT	0		MA	USA	42.3399	-71.06388888888889	0

Alternate Delivery: Visualization

HPCC Systems provides built-in Visualization of your output data in a variety of charts and graphs. You can visualize your data in three ways:

- Using the Chart Tool in the ECL Playground.
- Accessing the Visualize tab in all ECL workunits
- Using the Resources tab in conjunction with the ECL Visualizer bundle.

Installing:

ecl bundle install <https://github.com/hpcc-systems/Visualizer.git>

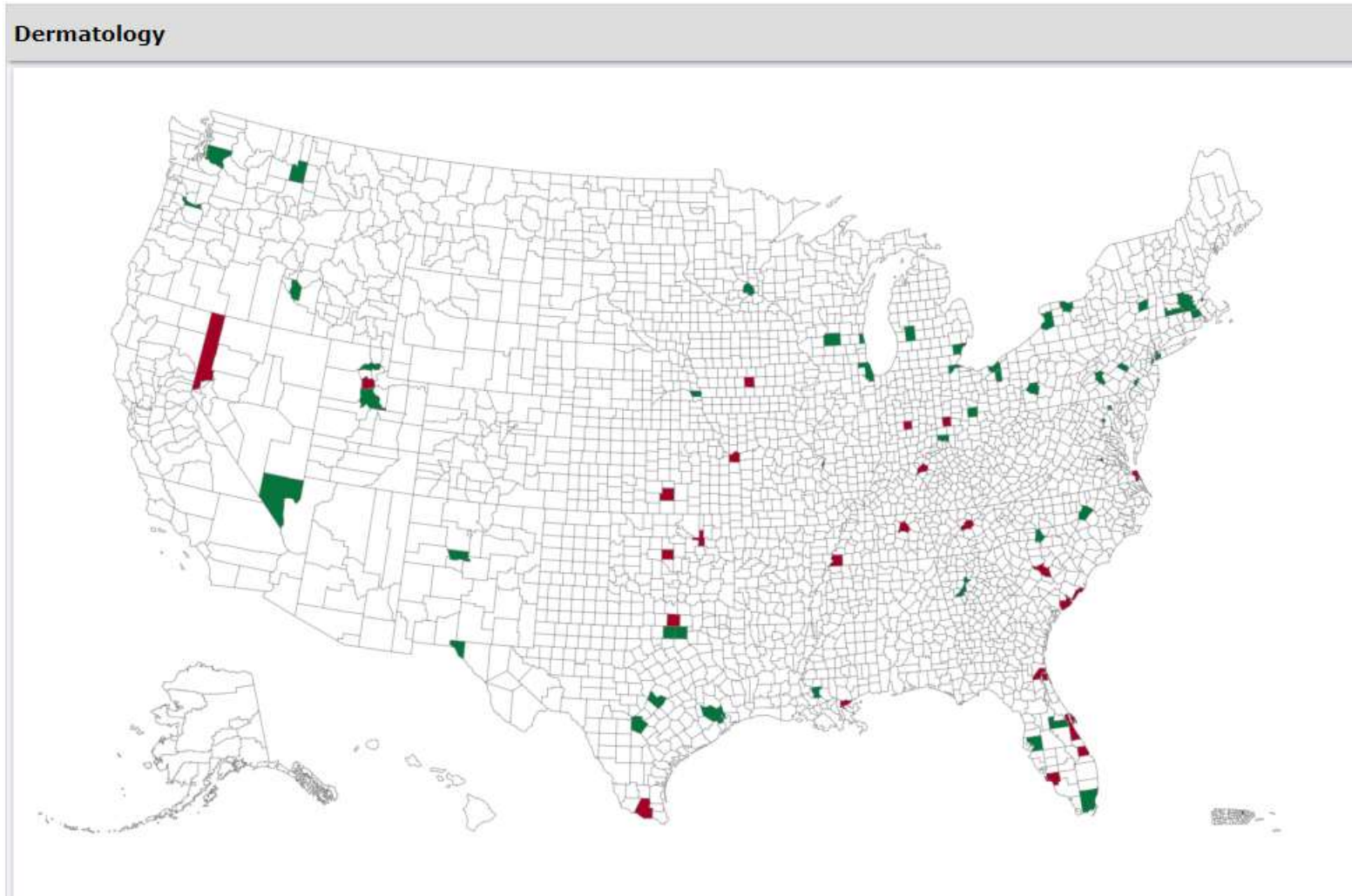
<https://hpccsystems.com/resources/visualizing-ecl-and-sharing-your-results-the-hpcc-systems-visualizer/>

<https://github.com/hpcc-systems/Visualizer>

Visualization Examples:

```
1  IMPORT $,Visualizer;  
2  
3  Cities := $.File_AllData.City_DS;  
4  
5  //Build Table  
6  DensityTbl := TABLE(Cities,{(INTEGER)county_fips,(INTEGER)density});  
7  
8  OUTPUT(DensityTbl,NAMED('DenFIPS'));  
9  
10 Visualizer.Choropleth.USCounties('Fips_demo',,'DenFIPS', , , DATASET([{'paletteID', 'Default;t'}], Visualizer.KeyValueDef));  
11  
12
```

Visualization Examples:



Resources!

Learn ECL Academy

<https://hpccsystems-solutions-lab.github.io>

ECL Training containing six short videos

https://www.youtube.com/watch?time_continue=192&v=Lk78BCctM-0

ECL Documentation

http://cdn.hpccsystems.com/releases/CE-Candidate-9.4.30/docs/EN_US/ECLLanguageReference_EN_US-9.4.30-1.pdf

Visualization Document

https://cdn.hpccsystems.com/releases/CE-Candidate-9.4.30/docs/EN_US/VisualizingECL_EN_US-9.4.30-1.pdf

Standard Library

https://cdn.hpccsystems.com/releases/CE-Candidate-9.4.30/docs/EN_US/ECLStandardLibraryReference_EN_US-9.4.30-1.pdf

Machine Learning

<https://hpccsystems.com/download/free-modules/machine-learning-library>

Thank You for watching!!



